IN THE SPECIFICATION:

Please replace the paragraph beginning at page 1, line 13, with the following rewritten paragraph:

-- The subject matter of this application is related to the subject matter in a co-pending non-provisional application by the same inventor as the instant application and filed on the same day as the instant application entitled. "Apparatus for Assisting Video Compression in a Computer System," having serial number ~~TO BE ASSIGNED~~ 09/048,932, and filing date March 26, 1998 ~~TO BE ASSIGNED~~ (Attorney Docket No. MEI-97-01386.00).--

Please replace the paragraph beginning at page 1, line 23, with the following rewritten paragraph:

-- The present invention ~~relates~~ relates to compressing video data, and more specifically to a method that provides assistance to a computer system in compressing a stream of video data on-the-fly, as the video data streams into the computer system.--

Please replace the paragraph beginning at page 2, line 5, with the following rewritten paragraph:

-- As video data is increasingly used in computer systems in applications such as video conferencing and video recording, computer systems often cannot keep pace with the computational requirements of video data. Video data streams typically have extremely large bandwidth ~~requires~~ requirements that can tax the capabilities of even the most high-speed processor to compress the video data for storage, or for transmission across a computer network or a telephone system. This compression is typically performed by a central processing unit (CPU) in a computer system with a resulting loss in image clarity

in image clarity due to the failure of the CPU to keep pace with the video data. Complex scenes, having many elements that are in motion represent the greatest challenge because they place a tremendous burden on the CPU during the compression and data transfer processes.--

Please replace the paragraph beginning at page 3, line 1, with the following rewritten paragraph:

-- What is needed is an apparatus or a method for off-loading the time-consuming task of computing the difference between successive frames of video data from the CPU of a computing system.--

Please replace the paragraph beginning at page 5, line 25, with the following rewritten paragraph:

-- FIG. 1 illustrates a computer system including a graphics controller with a difference ~~engine 106~~ engine in accordance with an embodiment of the present invention. The embodiment illustrated in FIG. 1 includes central processing unit (CPU) 120, which is coupled through north bridge 118 to memory 122 and bus 116. CPU 120 may be any type of central processing unit that can be used in a computer system. This includes, but is not limited to, a microprocessor CPU, a mainframe CPU and a device controller CPU. North bridge 118 forms part of the "core logic" for the computer system. This core logic ties together and coordinates operations of components in the computer system. Memory 122 can be any type of semiconductor memory that can be used in a computer system. Bus 116 can be any type of computer system bus. In one embodiment, bus 116 includes a PCI bus.--

Please replace the paragraph beginning at page 7, line 1, with the following rewritten paragraph:

--Video unit 102 receives video input 100 in analog form and converts it to digital form. In the illustrated embodiment, video unit 102 receives video input 100 in either PAL or NTSC format, and produces digital video data in YUV ~~format~~ data 104. Video unit 102 may include the BT829 chip produced by Rockwell Semiconductor Systems, Inc. of Newport Beach, California. Alternatively, the Rockwell BT848 part may be used to transfer data across a computer system bus into system memory or into a video controller's memory. (In some embodiments, these may be the same memory). Additionally, video data may be received from external sources through serial buses that can stream video data into system memory, usually by transferring data across bus 116. These serial buses may include the USB or the IEEE 1394 bus.--

Please replace the paragraph beginning at page 7, line 24, with the following rewritten paragraph:

-- FIG. 2 illustrates a computer system including a graphics controller incorporated into a core logic unit 200 in accordance with another embodiment of the present invention. This embodiment is similar to the embodiment illustrated in FIG. 1, except that graphics controller 106 and north bridge 118 from FIG. 1 are combined into a single core logic ~~unit~~ unit 200 with graphics ~~controller 200~~ controller. Additionally, memory 108 and memory 122 from FIG. 1 are combined into a single memory 122 in FIG. 2.--

Please replace the paragraph beginning at page 8, line 7, with the following rewritten paragraph:

--The embodiment illustrated in FIG. 2 operates in the same way as the embodiment illustrated in FIG. 1, except that in FIG. 2, unmodified video data 110 and XOR video data 112 are not stored in a separate graphics memory 108, but are rather stored in the system memory 122. Hence, CPU 120 does not have to reach out across bus 116 to retrieve XOR video data 112 from a separate graphics memory to complete the compression process. It merely has to retrieve ~~data~~ the XOR video data 112 from the system memory.--

Please replace the paragraph beginning at page 8, line 16, with the following rewritten paragraph:

--FIG. 3 illustrates the internal structure for a portion of a graphics controller that computes the difference between successive video frames in accordance with an embodiment of the present invention. The circuitry illustrated in FIG. 3 can exist in either graphics controller 106 from FIG. 1 or in core logic unit 200 from FIG. 2. The circuitry illustrated in FIG. 3 includes YUV data ~~input~~ 104, which feeds through color space conversion module 302. This module may perform color re-mapping on YUV data 104. The output of color space conversion module 302 feeds into video input buffer 304. From video input buffer 304, the video data feeds either into XOR unit 308 and multiplexer (MUX) 312. XOR unit 308 takes another input from previous frame buffer 306 and generates an output, which feeds into result buffer 310. Data from result buffer 310 feeds through MUX 312 and I/O buffers 316 into memory 108. MUX 312 takes another input from other write circuits 314. This allows data to be written to memory 122 from other sources. Data read from memory 122 feeds into previous frame buffer 306, and then into XOR unit 308. Alternatively, data read from

unit 308. Alternatively, data read from memory 122 may feed into other read circuits 315, allowing data to be read from memory 122 by other sources. Data read from memory 122 may also pass through serializer 330, color lookup table 332 and digital-to-analog converter 334 before becoming video output 114 to a monitor. Serializer 330 converts data read from memory 122 into a serial bitstream. This bitstream is modified in color lookup table 332, and is ultimately converted into analog form in digital-to-analog converter 334.--

Please replace the paragraph beginning at page 9, line 10, with the following rewritten paragraph:

--The circuitry illustrated in FIG. 3 operates as follows. Video data in YUV form 104 from video unit 102 streams into video input buffer 304 through color space conversion module 302. From video input buffer 304, this video data feeds through MUX 312 and I/O buffers 316 into unmodified video data 110 within memory 122. At the same time, data for a previous frame from unmodified video data 110 in memory 122 feeds into previous frame buffer 306 through I/O buffer 316. From previous frame buffer 306 this data feeds into XOR unit 308. XOR unit 308 computes the difference between data from the previous frame, stored in previous frame buffer 306, and data from the current frame, stored in video input buffer 304. The output of XOR unit 308 feeds into result buffer 310. From result ~~buffer 319~~ buffer 310, this data feeds through MUX 312 and I/O buffers 316 into an area for storing XOR video data 112 within memory 122. CPU 120 then uses this difference information to compress the video data.--

Please replace the paragraph beginning at page 10, line 9, with the following rewritten paragraph:

--FIG. 4 is a flow chart illustrating a method for compressing video data in a computer system in accordance with an embodiment of the present invention. This flow chart is divided into two columns. The column on the left-hand-side represents operations of the computational unit, and the column on the right-hand-side represents operations of the memory system. In this embodiment, the system starts in state 400. From state 400, the computational proceeds ~~state~~ to state 402. In state 402, the computational unit receives a stream of data from a current video frame from a video source. The computational unit next proceeds to state 404. In state 404, the computational unit performs a color space conversion on the video data. The computational unit next proceeds to state 406. In state 406, the computational unit computes a difference frame from a current video frame and a previous video frame received from the memory system "on-the-fly" as the current video frame streams into the computer system. In one embodiment, this difference computation takes place without intervention by the CPU 120. The computational unit next proceeds to state 412. In state 412, the computational unit produces compressed video data using the difference frame. The computational unit then loops back around to state 402 to process more video data.--